Red Hat Satellite 5.7
Proxy Installation Guide

Installing and configuring Red Hat Satellite Proxy Server

Red Hat Satellite Documentation Team

Red Hat Satellite 5.7 Proxy Installation Guide

Installing and configuring Red Hat Satellite Proxy Server

Red Hat Satellite Documentation Team

Legal Notice

Copyright © 2014 Red Hat.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document provides guidance on installing, configuring, and updating a Red Hat Satellite Proxy Server. For further information, see the Red Hat Satellite Getting Started Guide and the Red Hat Satellite Installation Guide.

Table of Contents

Preface **2**
1. Getting Help and Giving Feedback 2
1.1. Do You Need Help? 2
1.2. We Need Feedback 2

Chapter 1. Introduction to Red Hat Satellite Proxy **3**
1.1. Red Hat Satellite Proxy Server 3
1.2. Architecture and Operations 3

Chapter 2. Requirements **6**
2.1. Software Requirements 6
2.2. Hardware Requirements 7
2.3. Additional Requirements 8

Chapter 3. Installing Red Hat Satellite Proxy **10**
3.1. Summary of Installation Steps 10
3.2. Obtaining Satellite Proxy Entitlements 10
3.3. Uploading Satellite Proxy Entitlements to Satellite 11
3.4. Installing the Operating System on the Satellite Proxy Host 11
3.5. Installing the Red Hat Satellite Proxy Server Packages 12
3.6. Running the Red Hat Satellite Proxy Installation Script 13
3.7. Performing Post-installation Tasks 14
3.8. Automating Satellite Proxy Server Installation 16

Chapter 4. Custom Channel Package Manager **18**
4.1. Using the Custom Channel Package Manager and Serving Local Packages through the Red Hat Network Proxy 18
4.1.1. Creating a Private Channel 18
4.1.2. Uploading Packages 19
4.2. Configuring Proxy Precaching 20
4.2.1. Manually Loading RPM Files into the Proxy Cache 20
4.2.2. Automatically Loading RPM Files into the Proxy Cache 20

Chapter 5. Configuring Satellite Proxy to Use CNAME Records **22**
5.1. Prerequisites 22
5.2. Adding CNAME Records to the Satellite Proxy Server Configuration 22
5.3. Generating and Using Multi-host SSL Certificates 23

Chapter 6. Load Balancing Satellite Proxy Servers **25**
6.1. Installing a Squid Reverse Proxy 25
6.2. Setting up the Client 27
6.3. Testing the Configuration 27

Chapter 7. Upgrading a Red Hat Proxy Server Installation **29**
7.1. Prerequisites 29
7.2. Upgrading Your Proxy Installation 29

Sample Satellite Proxy Server Configuration File **31**

Glossary of Terms **32**

Revision History **33**

Preface

1. Getting Help and Giving Feedback

1.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com. From the Customer Portal, you can:

- Search or browse through a knowledge base of technical support articles about Red Hat products.
- Submit a support case to Red Hat Global Support Services (GSS).
- Access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click the name of any mailing list to subscribe to that list or to access the list archives.

1.2. We Need Feedback

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product Red Hat Satellite.

When submitting a bug report, be sure to mention the manual's identifier: *Proxy_Installation_Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Introduction to Red Hat Satellite Proxy

1.1. Red Hat Satellite Proxy Server

Red Hat Satellite Proxy Server is a package-caching mechanism that reduces the bandwidth requirements for Red Hat Satellite and enables custom package deployment. Satellite Proxy customers cache RPM packages, such as Errata Updates from Red Hat or custom packages generated by their organization, on an internal, centrally-located server. Client systems then receive these updates from Red Hat Satellite Proxy rather than by accessing the Internet individually.

Although the packages are served by Red Hat Satellite Proxy, clients' system profiles and user information are stored on a secure, central Red Hat Satellite Server. The Satellite Proxy acts as a go-between for client systems and the Red Hat Satellite Server. Only the package files are stored on the Satellite Proxy. Every transaction is authenticated, and the **Red Hat Update Agent** checks the GPG signature of each package retrieved from the local Satellite Proxy.

In addition to storing official Red Hat packages, the Satellite Proxy Server can be configured to deliver an organization's own custom packages from private *channels*. For example, an organization could develop its own software, package it in an RPM, sign it with its own GPG signature, and have the local Satellite Proxy Server update all of the individual systems in the network with the latest versions of the custom software.

Advantages of using Satellite Proxy Server include:

- Scalability: one organization can support multiple local Red Hat Satellite Proxies.
- Security: a secure connection is maintained from the client systems to the local Satellite Proxy, and from there to the Red Hat Satellite servers.
- Saves time: packages are delivered significantly faster over a local area network than the Internet.
- Saves bandwidth: packages are only downloaded once from Red Hat Satellite (using the local Satellite Proxy Server's caching mechanism), instead of downloading each package separately to each client system.
- Customized updates: create an automated package delivery system for custom software packages, as well as official Red Hat packages required for the client systems. Customized, private Red Hat Satellite channels allow an organization to automate delivery of in-house packages.
- Customized configuration: restrict or grant updates to specific architectures and operating system versions.

Report a bug

1.2. Architecture and Operations

The Red Hat Update Agent or **Package Updater** on the client systems does not directly contact a Red Hat Satellite Server. Instead, the client (or clients) connects in turn to a Satellite Proxy Server that connects to a Red Hat Satellite Server. Thus, the client systems do not need direct access to the Internet. They need access only to the Satellite Proxy Server.

Important

Red Hat strongly recommends that clients connected to a Satellite Proxy server be running the latest update of Red Hat Enterprise Linux to ensure proper connectivity.

Clients that access a Red Hat Satellite Proxy are still authenticated by Red Hat Satellite but in this case the Satellite Proxy provides both authentication and route information to Red Hat Satellite. After a successful authentication, the Red Hat Satellite Server informs the Satellite Proxy server that it is permitted to execute a specific action for the client. The Satellite Proxy server downloads all of the updated packages (if they are not already present in its cache) and delivers them to the client system.

Requests from the Red Hat Update Agent or Package Updater on the client systems are still authenticated on the server side, but package delivery is significantly faster because the packages are cached in the HTTP Proxy Caching Server or the Satellite Proxy server (for local packages). The Satellite Proxy server and client system are connected over the LAN and transfer speeds are limited only by the speed of the local network.

The authentication process proceeds as follows:

1. The client performs a login action at the beginning of a client session. This login is passed through one or more Satellite Proxy Servers until it reaches a Red Hat Satellite server.

2. The Red Hat Satellite server attempts to authenticate the client. If authentication is successful, the server returns a *session token* through the chain of Satellite Proxy Servers. This token, which has a signature and expiration time, contains user information, such as channel subscriptions, username, and so on.

3. Each Satellite Proxy Server caches this token on its local file system in the **`/var/cache/rhn/`** directory. Caching reduces some of the overhead of authenticating with Red Hat Satellite servers and greatly improves the performance of Red Hat Satellite.

4. This session token is sent to the client machine and is used in subsequent actions on Red Hat Satellite.

From the client's perspective, there is no difference between a Satellite Proxy Server and a Red Hat Satellite server. From the Red Hat Satellite server's perspective, a Satellite Proxy Server is a special type of Red Hat client. Clients are thus not affected by the route that a request takes to reach a Red Hat Satellite server. All of the logic is implemented in the Satellite Proxy Servers and Red Hat Satellite servers.

The Custom Channel Package Manager can also be installed and configured to serve custom packages. Any package that is not an official Red Hat package, including custom packages written specifically for an organization, can only be served from a private software channel (also referred to as a custom software channel). After creating a private Red Hat Satellite channel, the custom RPM packages are associated with that channel by uploading the package headers to the Red Hat Satellite servers. Only the headers are uploaded, not the actual package files. The headers are required because they contain crucial RPM information, such as software dependencies, that allows Red Hat Satellite to automate package installation. The actual custom RPM packages are stored on the Satellite Proxy Server and sent to the client systems from inside the organization's local area network.

Configuring a computer network to use Satellite Proxy Servers is straightforward. The Red Hat Satellite applications on the client systems must be configured to connect to the Satellite Proxy Server instead of a Red Hat Satellite server. See the *Client Configuration Guide* for details. On the proxy side, you need to specify the next proxy in the chain (which eventually ends with a Red Hat Satellite

server). If the Red Hat Package Manager is used, the client systems must be subscribed to the private Red Hat channel.

Report a bug

Chapter 2. Requirements

This chapter focuses on the prerequisites for installing Red Hat Satellite Proxy Server.

2.1. Software Requirements

To perform an installation, the following software-related components must be available:

- Base operating system: Satellite 5.7 and Satellite Proxy 5.7 are only supported on Red Hat Enterprise Linux 6. You can install the operating system using any of the methods supported by Red Hat.
- Satellite Proxy requires an equal or high version of Satellite. For example, Satellite Proxy 5.6 works with Satellite 5.6 and Satellite 5.7, but Satellite Proxy 5.7 works only with Satellite 5.7.
- A Satellite Proxy entitlement within the Satellite Server account.
- A Provisioning entitlement within the Satellite Server account (this is packaged with the Satellite Proxy entitlement).
- Access to the Red Hat Network Tools channel for the installed version of Red Hat Enterprise Linux. This channel includes the *spacewalk-proxy-installer* package that contains the **`configure-proxy.sh`** installation program required to install Satellite Proxy.
- All *rhncfg** packages installed on the Satellite Proxy (from the Red Hat Network Tools channel).
- Either the *spacewalk-certs-tools* package installed on the Satellite Proxy (from the Red Hat Network Tools channel) for Hosted users, or the secure sockets layer (SSL) CA certificate password used to generate the parent server certificate for Satellite Server users.

Important

Red Hat Satellite Proxy Server 5.7 can only run on Red Hat Enterprise Linux 6. It can manage clients of any RHEL version when used with Satellite 5.7. Satellite Proxy Server 5.7 cannot manage Red Hat Enterprise Linux 7 clients in a Proxy-only environment.

Virtualization Support

You can install Satellite Proxy on Red Hat Enterprise Linux 5 or 6 in any virtualized environment that Red Hat supports, including Xen, Hyper-V, KVM, and VMware. For more information about supported environments, see https://access.redhat.com/site/certified-hypervisors.

In production deployments, Red Hat recommends that you deploy Satellite Proxy as the sole application running on the underlying physical hardware to avoid contention issues. Also, be aware that functional support for virtualized environments does not always equal the performance of running on physical hardware, so carefully consider the virtualized environment of choice and any recommended tuning guidelines.

Each purchased Satellite Proxy includes one supported instance of Red Hat Enterprise Linux Server. Satellite Proxy must be installed onto a fresh installation of Enterprise Linux, and where Satellite Proxy is the only application and service provided by the operating system. Using the Red Hat Enterprise Linux operating system included in Satellite Proxy to run other daemons, applications, or services within the environment is not supported.

Obtaining the Required Package Sets

Each version of Red Hat Enterprise Linux requires a certain package set to support Satellite Proxy. Adding more packages can cause errors during installation. Therefore, Red Hat recommends obtaining the desired package set in the following ways:

- For kickstart installations, specify the following package group: **@Base**
- For installing Red Hat Enterprise Linux from CD or ISO image, select the following package group: **Minimal**

Report a bug

2.2. Hardware Requirements

Use the following guidelines to determine if your hardware is suitable for a Satellite Proxy installation:

- A Pentium IV Processor or equivalent.
- At least 2 GB of memory. 4 GB of memory recommended.
- At least 5 GB of storage for a base installation of Red Hat Enterprise Linux.
- 6 GB of storage per architecture per release.

  The caching mechanism used by Satellite Proxy is the Squid HTTP proxy, which saves significant bandwidth for the clients. Cached packages are stored in the **/var/spool/squid** directory. The more disk space that the cache has available, the less likely it is to remove RPM files before they have expired. Providing less than the recommended storage reduces the performance enhancements offered by using the Proxy server.

Satellite Proxy 5.7 features the ability to *precache* entire channels. Using this feature requires significant disk space; Red Hat recommends at least 20 GB per base channel. Refer to the hardware guidelines in the *Red Hat Satellite Installation Guide* in you intend to use this feature. Proxy precaching is discussed in Section 4.2, “Configuring Proxy Precaching”.

If the Satellite Proxy is configured to distribute custom, or local packages, make sure that the **/var** mount point on the system storing local packages has sufficient disk space to hold all of the custom packages, which are stored in **/var/spool/rhn-proxy**. The required disk space for local packages depends on the number of custom packages served.

The load on the Apache Web server is directly related to the frequency with which client systems connect to the Satellite Proxy. If the default interval of four hours (or 240 minutes) is reduced, the load on this component *increases* significantly. This value is set in the **/etc/sysconfig/rhn/rhnsd** configuration file of each client system.

Note

The installation procedure optimizes the caching abilities for the available resources. Provide more memory and disk space prior to installation to increase active caching within Red Hat Satellite Proxy.

Report a bug

2.3. Additional Requirements

2.3. Additional Requirements

The following additional requirements must be met before the Satellite Proxy installation can be considered complete:

Full Access

Client systems need full network access to the Satellite Proxy services and ports.

Firewall Rules

Red Hat strongly recommends setting up a firewall between the Satellite Proxy and the Internet. However, depending on your Satellite Proxy implementation, you need to open several TCP ports in this firewall:

Table 2.1. Ports to Open on the Satellite Proxy

Port	Direction	Reason
80	Outbound	The Satellite Proxy uses this port to reach your Satellite URL.
80	Inbound	Client requests arrive using either HTTP or HTTPS.
443	Inbound	Client requests arrive using either HTTP or HTTPS.
443	Outbound	The Satellite Proxy uses this port to reach the Satellite URL.
4545	Outbound	If your Satellite Proxy is connected to a Satellite Server, Monitoring makes connections to **rhnmd** running on client systems through this TCP port, if Monitoring is enabled and probes are configured to registered systems.
5222	Inbound	Allows **osad** client connections to the **jabberd** daemon on the Satellite Proxy when using Red Hat Network Push technology.
5269	Inbound and Outbound	If the Satellite Proxy is connected a Satellite Server, this port must be open to allow server-to-server connections using **jabberd** for Red Hat Network Push Technology.

Synchronized System Times

Time sensitivity is a significant factor when connecting to a Web server running SSL (Secure Sockets Layer); it is imperative the time settings on the clients and server are close together so that the SSL certificate does not expire before or during use. It is recommended that Network Time Protocol (NTP) be used to synchronize the clocks.

Fully Qualified Domain Name (FQDN)

The system upon which the Satellite Proxy is installed must resolve its own FQDN properly.

Distribution Locations

Because the Satellite Proxy forwards virtually all local HTTP requests to Red Hat Satellite, take care in putting files destined for distribution (such as in a kickstart installation tree) in the non-forwarding location on the Satellite Proxy: **/var/www/html/pub/**. Files placed in this directory can be downloaded directly from the Satellite Proxy. This can be especially useful for distributing GPG keys or establishing installation trees for kickstart files.

Bandwidth

Network bandwith is important for communication among Satellites, Proxies, and Clients. To accomodate high volume traffic, Red Hat recommends a high bandwidth on a network capable of delivering packages to many systems and clients. As a guide, Red Hat provides a set of estimates for package transfer from one system to another over various speeds.

Table 2.2. Bandwidth estimates

	Single Package (10Mb)	Minor Release (750Mb)	Major Release (6Gb)
256Kbps	5 Mins 27 Secs	6 Hrs 49 Mins 36 Secs	2 Days 7 Hrs 55 Mins
512Kbps	2 Mins 43.84 Secs	3 Hrs 24 Mins 48 Secs	1 Day 3 Hrs 57 Mins
T1 (1.5Mbps)	54.33 Secs	1 Hr 7 Mins 54.78 Secs	9 Hrs 16 Mins 20.57 Secs
10Mbps	8.39 Secs	10 Mins 29.15 Secs	1 Hr 25 Mins 53.96 Secs
100Mbps	0.84 Secs	1 Min 2.91 Secs	8 Mins 35.4 Secs
1000Mbps	0.08 Secs	6.29 Secs	51.54 Secs

Red Hat recommends at least a 100Mbps network speed for minor and major releases. This avoids timeouts for transfers longer than 10 minutes. All speeds are relative to your network setup.

Red Hat recommends that the system running the code should not be publicly available. Only system administrators should have shell access to these machines. All unnecessary services should be disabled. Use **`ntsysv`** or **`chkconfig`** to disable services.

Report a bug

Chapter 3. Installing Red Hat Satellite Proxy

This chapter describes the installation and basic configuration of Red Hat Satellite Proxy Server. It assumes the prerequisites listed in Chapter 2, *Requirements* have been met. However, if you are *upgrading* to a later version of Red Hat Satellite Proxy Server, see Chapter 7, *Upgrading a Red Hat Proxy Server Installation*.

3.1. Summary of Installation Steps

A functional Red Hat Satellite Proxy requires more than installing software. Client systems require configuration, and the Satellite Proxy often requires that custom repositories be set up. This section provides a high-level list of steps.

1. Obtain and install the Red Hat Satellite Proxy entitlements on your Red Hat Satellite server. This involves creating an updated entitlement certificate on the Red Hat Customer Portal. See Section 3.2, “Obtaining Satellite Proxy Entitlements”.
2. Upload the updated entitlement certificate on your Red Hat Satellite server and synchronize the required channels for Satellite Proxy. See Section 3.3, “Uploading Satellite Proxy Entitlements to Satellite”.
3. Create or provision a new Red Hat Enterprise Linux system for use as the Satellite Proxy host and register it to Satellite. See Section 3.4, “Installing the Operating System on the Satellite Proxy Host”
4. Install the required packages on the Satellite Proxy host. See Section 3.5, “Installing the Red Hat Satellite Proxy Server Packages”
5. Run the **`configure-proxy.sh`** installation script. See Section 3.6, “Running the Red Hat Satellite Proxy Installation Script”
6. Perform any necessary post-configuration tasks. See Section 3.7, “Performing Post-installation Tasks”

3.2. Obtaining Satellite Proxy Entitlements

Satellite Proxy Server is a package-based installation through the Satellite server. The first step involves adding the Proxy entitlements to your Satellite server. This requires an updates entitlement certificate, which you generate with the following steps:

1. Navigate to access.redhat.com in your web browser.
2. Log in using your Red Hat customer account details.
3. Navigate to **`Subscriptions`**.
4. Scroll to the **`Manage`** section and click **`Subscription Management Applications`**.
5. Select the **`Satellite`** tab.
6. Select the **`Name`** of your Satellite.
7. Click the **`Attach a subscription`** link and add the subscription containing the *RHN Tools for RHEL* channel to your entitlement certificate. Use the checkboxes to select the subscription type and use the **`Quantity`** dropdown selector to choose the number of subscriptions to add, which corresponds to the number of proxies to create. Click **`Attached Selected`** to

add these subscriptions to the entitlement certificate.

8. Click the **Download Satellite Certificate** and save the entitlement certificate.

The updated certificate now contains entitlements for Red Hat Satellite Proxy. The next section shows how to upload this refreshed certificate to your Satellite server.

3.3. Uploading Satellite Proxy Entitlements to Satellite

The next step requires uploading the updating entitlements certificate to your Satellite server, which involves the following steps:

1. Log into your Red Hat Satellite server as an administration user.
2. Navigate to **Admin** → **Red Hat Satellite Configuration** → **Certificate**.
3. Click **Browse** and select the updated certificate file.
4. Synchronize the required Proxy channel and RHN Tools channel. Log in through SSH to the Satellite server and run the following command as **root**:

```
[root@satellite ~]# satellite-sync -c rhn-tools-rhel-x86_64-server-6 -c redhat-rhn-proxy-5.7-server-x86_64-6
```

5. Once synchronized, add these entitlments to an activation key. Navigate to **Systems** → **Activation Keys** and click **create new key**. Add the base channel for these entitlement, such as Red Hat Enterprise Linux base channel, and select **Provisioning** and **Monitoring** entitlements. Click **Create Activation Key** to complete.

The Satellite server now contains the required entitlements to install the Red Hat Satellite Proxy. The resulting activation key contains these entitlements and is used to register the Proxy host to the required channels.

3.4. Installing the Operating System on the Satellite Proxy Host

Red Hat Satellite Proxy Server is designed to run on the Red Hat Enterprise Linux operating system. The first step is to install the base operating system. See the Red Hat Enterprise Linux 6 Installation Guide [1] for details.

Note

If you have base channels for Red Hat Enterprise Linux 6 on your Satellite server, you can kickstart the machine to install Red Hat Enterprise Linux 6. See the *Red Hat Satellite Getting Started Guide* for more information on kickstarts.

During and after the operating system installation process, ensure that you perform the following tasks:

- Allocate sufficient space to the partition used to store packages, according to the hardware requirements prescribed earlier. The default location for cached Red Hat packages is **/var/spool/squid**; custom packages are located in **/var/spool/rhn-proxy**.

Note

The installation program automatically calculates the available space on the partition where **/var/spool/squid** is mounted and allocates up to 60 per cent of the free space for use by Satellite Proxy.

- Ensure the firewall rules are updated to meet the requirements stated in Section 2.3, "Additional Requirements". Modify your **iptables** settings and restart the service.
- Install the packages required by Red Hat Satellite Proxy Server.

Important

Install only the base Satellite Proxy packages as installing additional packages might cause the Satellite Proxy installation to fail.

See Section 2.1, "Software Requirements" for how to obtain the correct package group needed for each version of Red Hat Enterprise Linux.

- Enable **Network Time Protocol (NTP)** and select the appropriate time zone on the Satellite Proxy and on all client systems.

After completing the installation of the base operating system, log in to the Satellite Proxy host as the root user and register the system to the Red Hat Satellite Server using the **rhnreg_ks** command. For example:

```
[root@satproxy ~]# rhnreg_ks --serverUrl=http://sat57.example.com/XMLRPC
--activationkey=1-123456789abcedf123456789abcdef12
```

3.5. Installing the Red Hat Satellite Proxy Server Packages

The following instructions describe the Satellite Proxy installation process:

1. Log in as the root user on the intended Satellite Proxy system.
2. Subscribe the client to the required channels:

   ```
   [root@satproxy ~]# spacewalk-channel --add -c rhn-tools-rhel-
   x86_64-server-6 --user admin --password adminpassword
   ```

3. Assign a provisioning entitlement to the system:
   a. On the Satellite server, click on **Systems**.
   b. Click the Satellite Proxy's name.
   c. Click the **Properties** tab.
   d. Choose **Provisioning** in the **Add-On Entitlements** section.
   e. Click **Update Properties**.

4. Install the Satellite Proxy installation package. This package contains the main script that leads you through the actual Satellite Proxy installation.

```
[root@satproxy ~]# yum install spacewalk-proxy-installer
```

3.6. Running the Red Hat Satellite Proxy Installation Script

The command-line installation program guides you through the actual Satellite Proxy installation process. This program presents a series of prompts regarding Satellite Proxy installation and initial configuration details, such as installation options and SSL certificate generation. You need root access to the server to perform this step.

Important

Before you run the installation script, the Satellite Proxy requires the SSL files from your Satellite server. Create the **/root/ssl-build** directory:

```
[root@satproxy ~]# mkdir /root/ssl-build
```

Copy the public certificate and CA certificate from the desired Red Hat Satellite to the new directory:

```
[root@satproxy ~]# scp 'root@sat57.example.com:/root/ssl-build/{RHN-
ORG-PRIVATE-SSL-KEY,RHN-ORG-TRUSTED-SSL-CERT,rhn-ca-openssl.cnf}'
/root/ssl-build
```

Alternatively, append the **--force-own-ca** option when you run the installation script.

Run the following installation script to install Red Hat Satellite Proxy Server:

```
# configure-proxy.sh
```

Note

You can press **Enter** at any prompt to use the default response enclosed in brackets. Alternatively, use the **--non-interactive** option with the installation script if you want to use default answers without *any* user interaction.

Gathering Satellite Proxy Server Installation Information

The installation script requests details about the Satellite Proxy installation specific to the machine where you are performing the installation.

RHN Parent

The Satellite Server domain name or address of the system that serves content to the Satellite Proxy.

CA Chain

Press **Enter** to use the default path for the Certificate Authority (CA) Chain.

If the Satellite Proxy is communicating with Red Hat Satellite, then this value is usually **/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT**. Otherwise, custom SSL certificates must be located in the **/usr/share/rhn/** directory.

Satellite Proxy version to activate

Request for confirmation of the version of Satellite Proxy to install.

HTTP Proxy

If the Satellite Proxy server connects through an HTTP proxy, enter the proxy host name and port number, for example, *corporate.proxy.example.com:3128*

Traceback email

A comma-separated list of email addresses to which error-related traceback messages are sent, usually the email of the Satellite Proxy administrator.

Use SSL

Press **Enter** or type **y** to configure the Satellite Proxy to use SSL.

SSL Certificate

Enter the details necessary to generate a valid SSL server certificate. Example 3.1, “Example Generation of an SSL Certificate” demonstrates generating such a certificate.

Example 3.1. Example Generation of an SSL Certificate

```
Regardless of whether you enabled SSL for the connection to the
Satellite Proxy Parent
Server, you will be prompted to generate an SSL certificate.
This SSL certificate will allow client systems to connect to
this Satellite Proxy
securely. See the Satellite Proxy Installation Guide for more
information.
Organization: Example Company
Organization Unit [proxy1.example.com]:
Common Name: proxy1.example.com
City: New York
State: New York
Country code: US
Email [admin@example.com]:
```

The installation script installs multiple sets of packages and will ask for your permission to install them. Select **Y** at each prompt.

3.7. Performing Post-installation Tasks

After the main Satellite Proxy installation tasks have been completed, the installation program performs a number of additional tasks, such as prompting for installation of monitoring support, creating configuration channels, and restarting modified daemons.

Install Monitoring support

```
You do not have monitoring installed. Do you want to install it?
Will run 'yum install spacewalk-proxy-monitoring'. [Y/n]:
```

Confirm whether or not you want to install Monitoring support on the Satellite Proxy server. This installs the Monitoring packages to the Satellite Proxy.

Configure SSL

The **configure-proxy.sh** program configures SSL, and prompts you to create a Certificate Authority password and confirm it before generating the SSL keys and the public certificate.

Example 3.2. Example Generation of CA Key and Public Certificate

```
Generating CA key and public certificate:
CA password: *********
CA password confirmation: *********
```

Create Configuration Channel

The installation program also requests confirmation that you want to create a configuration channel based on the configuration files created while running **configure-proxy.sh.**

The installation program creates a Satellite Server configuration channel based on the name of the system (the **sysID**) where the Satellite Proxy is installed (in the following example, the **sysID** is 1000010000), and collects the various **httpd, SSL, squid,** and **jabberd** server files that will comprise the configuration channel for the Satellite Proxy server.

An example of this configuration is shown in Example 3.3, “Example of Creating a Configuration Channel”.

Example 3.3. Example of Creating a Configuration Channel

```
Create and populate configuration channel
rhn_proxy_config_1000010000? [Y]:
Using server name satellite.example.com
Red Hat Network username: admin
Password:
Creating config channel rhn_proxy_config_1000010000
Config channel rhn_proxy_config_1000010000 created
using server name satserver.example.com
Pushing to channel rhn_proxy_config_1000010001:
Local file /etc/httpd/conf.d/ssl.conf -> remote file
/etc/httpd/conf.d/ssl.conf
Local file /etc/rhn/rhn.conf -> remote file /etc/rhn/rhn.conf
Local file /etc/rhn/cluster.ini -> remote file
/etc/rhn/cluster.ini
Local file /etc/squid/squid.conf -> remote file
/etc/squid/squid.conf
Local file /etc/httpd/conf.d/cobbler-proxy.conf -> remote file
/etc/httpd/conf.d/cobbler-proxy.conf
Local file /etc/httpd/conf/httpd.conf -> remote file
/etc/httpd/conf/httpd.conf
```

```
Local file /etc/jabberd/c2s.xml -> remote file
/etc/jabberd/c2s.xml
Local file /etc/jabberd/sm.xml -> remote file
/etc/jabberd/sm.xml
```

Restart services

The final step of the installation process is to restart all of the Satellite Proxy-related services. The installation program exits when this step is completed.

Example 3.4. Restarting all Satellite Proxy Server-related Services

```
Enabling Satellite Proxy
Shutting down rhn-proxy...
Shutting down Jabber router:                               [
OK  ]
Stopping httpd:                                            [
OK  ]
Stopping squid:                                            [
OK  ]
Done.
Starting rhn-proxy...
init_cache_dir /var/spool/squid... Starting squid: .       [
OK  ]
Starting httpd:                                            [
OK  ]
Starting Jabber services                                   [
OK  ]
Done.
```

If all services start successfully, the Satellite Proxy installation has been successful.

3.8. Automating Satellite Proxy Server Installation

To automate some of the process of installing Satellite Proxy on your systems, the **configure-proxy.sh** script allows administrators to create *answer files* that contain predetermined responses to prompts in the installation program.

The following is an example answer file that contains predetermined answers related to version number, the Satellite Server that serves as the parent server, **SSL,** and other configuration parameters. See the **configure-proxy.sh** manual page (**man configure-proxy.sh**) for more information about creating and using answer files.

```
# example of answer file for configure-proxy.sh
# for full list of possible option see
# man configure-proxy.sh

VERSION=5.7
RHN_PARENT=rhn-satellite.example.com
TRACEBACK_EMAIL=jsmith@example.com
USE_SSL=1
SSL_ORG="Red Hat"
SSL_ORGUNIT="Satellite"
```

```
SSL_CITY=Raleigh
SSL_STATE=NC
SSL_COUNTRY=US
INSTALL_MONITORING=N
ENABLE_SCOUT=N
CA_CHAIN=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
POPULATE_CONFIG_CHANNEL=Y
```

Use the **--answer-file** option with the **configure-proxy.sh** script to use an answer file to help automate your Satellite Proxy installation, as shown in the following example. Replace the example **answers_file.txt** file name with the path to your answer file.

```
# configure-proxy.sh --answer-file=/path/to/answers_file.txt
```

[1] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html-single/Installation_Guide/index.html

Chapter 4. Custom Channel Package Manager

This is a section on using Red Hat Satellite Proxy with the Custom Channel Package Manager.

4.1. Using the Custom Channel Package Manager and Serving Local Packages through the Red Hat Network Proxy

The Custom Channel Package Manager is a command line tool that allows an organization to serve local packages associated with a private Red Hat Satellite channel through the Red Hat Satellite Proxy Server. To update only official Red Hat packages for the Red Hat Satellite Proxy Server, do not install the Custom Channel Package Manager.

To use the Custom Channel Package Manager, install the **spacewalk-proxy-package-manager** package and its dependencies.

Only the header information for packages is uploaded to the Red Hat Satellite Servers. The headers are required so that Red Hat Satellite can resolve package dependencies for the client systems. The actual package files (***.rpm**) are stored on the Red Hat Satellite Proxy Server.

The Custom Channel Package Manager uses the same settings as the Proxy, defined in the **/etc/rhn/rhn.conf** configuration file. See the **rhn_package_manager** manual page for more information.

In order for Custom Channel Package Manager to be able to serve the local packages, the following steps need to be followed:

1. Create a private channel.
2. Upload the local packages into the channel.

The steps will be further discussed in the next sections.

4.1.1. Creating a Private Channel

Before local packages can be provided through the Red Hat Satellite Proxy Server, a private channel is needed to store them. Perform the following steps to create a private channel:

1. Log in to your local Red Hat Satellite server in the network.
2. Click **Channels** on the top navigation bar. If the **Manage Channels** option is not present in the left navigation bar, ensure that this user has channel editing permissions set. Do this through the **Users** category accessible through the top navigation bar.
3. In the left navigation bar, click **Manage Software Channels** and then the **create new channel** button at the top-right corner of the page.
4. Select a parent channel and base channel architecture, then enter a name, label, summary, and description for the new private channel. The channel label must: be at least six characters long, begin with a letter, and contain only lowercase letters, digits, dashes (-), and periods(.). Also enter the URL of the channel's GPG key. Although this field is not required, it is recommended to enhance security.
5. Click **Create Channel**.

Report a bug

4.1.2. Uploading Packages

Note

Only Organization Administrators can upload packages to private Red Hat Satellite channels. The script will prompt you for your Red Hat Satellite credentials.

After creating the private channel, upload the package headers for the binary and source RPMs to the Red Hat Satellite Server and copy the packages to the Red Hat Satellite Proxy Broker Server. To upload the package headers for the binary RPMs, issue the following command:

```
rhn_package_manager -c "label_of_private_channel" pkg-list
```

This command will upload the header of the package to the channel name specified, and the package itself to **/var/spool/rhn-proxy/rhn**.

pkg-list is the list of packages to be uploaded. Alternatively, use the **-d** option to specify the local directory that contains the packages to add to the channel. Ensure that the directory contains only the packages to be included and no other files. Custom Channel Package Manager can also read the list of packages from standard input (using **--stdin**).

To upload the package headers for the source RPMs:

```
rhn_package_manager -c "label_of_private_channel" --source pkg-list
```

If you have more than one channel specified (using **-c** or **--channel**), the uploaded package headers will be linked to all the channels listed.

Note

If a channel name is not specified, the packages are not added to any channel. The packages can then be added to a channel using the Red Hat Satellite web interface. The interface can also be used to modify existing private channels.

After uploading the packages, you can immediately check the Red Hat Satellite Web interface to verify their presence. Click **Channels** in the top navigation bar, **Manage Software Channels** in the left navigation bar, and then the name of the custom channel. Then click the **Packages** subtab. Each RPM should be listed.

Also check to see if the local directory is in sync with the Red Hat Satellite Server's image of the channels at the command line:

```
rhn_package_manager -s -c "label_of_private_channel"
```

The **-s** option will list all the missing packages (packages uploaded to the Red Hat Satellite Server not present in the local directory). You must be an Organization Administrator to use this command. The script will prompt you for your Red Hat Satellite username and password.

If you are using the Custom Channel Package Manager to update local packages, you must go to the Red Hat Satellite website to subscribe the system to the private channel.

4.2. Configuring Proxy Precaching

Your Proxy server can precache or mirror custom RPM files. This means that RPM files are delivered directly from the Proxy server to the clients; the clients do not have to wait for the files to download from the Satellite server to the Proxy server, and then be delivered to the client. The Proxy server recognizes RPM requests from **yum** as well as **anaconda** (for kickstart installations and provisioning). See the **rhn_package_manager** manual page for more information.

Proxy precaching is especially useful if the network connection to the Satellite server is slow or if bandwidth is at a premium. You can use the **rhn_package_manager** command to manually load RPM files into the Proxy server's cache, or you can create a **cron** job that uses the **rsync** command to perform the task automatically.

Note

Using the Proxy server precache feature requires that disk space be available at all times for the required RPM files. Unlike a non-precached Proxy server, where only requested RPM files exist, and only until they expire, precached RPM files remain indefinitely on the Proxy server whether they are used or not.

4.2.1. Manually Loading RPM Files into the Proxy Cache

Satellite Proxy and **rhn_package_manager** have been updated to avoid unwanted cache collisions. You can use the existing **rhn_package_manager --copyonly** command to populate the cache; (an alias to that option has been added with the more user-friendly name **--cache-locally**). Another significant change to **rhn_package_manager** is that it can now read and import packages from a channel export, which could for example be created on the Satellite server using the **rhn-satellite-exporter** utility. This is in addition to the other methods that **rhn_package_manager** can use to import RPM files, such as the **--dir** option for importing all RPM files in a directory, or a list of RPM files supplied on the command line.

The following example demonstrates how to cache only the RPM files that exist in the *my-channel-l* channel, using a channel export from the Satellite server. This export contains all the channels from the Satellite server, and is mounted on **/mnt/export**:

```
# rhn_package_manager --cache-locally --from-export /mnt/export --
channel my-channel-1
```

To import all the RPM files from all channels that the export contains, omit the **--channel** option:

```
# rhn_package_manager --cache-locally --from-export /mnt/export
```

If the channel export is spread across multiple ISO images it is not necessary to combine them locally on the Proxy before running the **rhn_package_manager** command. Mount the images one at a time and run the same command on each.

4.2.2. Automatically Loading RPM Files into the Proxy Cache

Populating the initial clone may be convenient and fast through a channel export or directory that contains RPM files, but it is inconvenient to have to create a new channel export or directory containing new RPM files every time the Satellite server is updated with new content. You can set up a

cron job or similar with an **rsync** command to download any updated RPM files to the Proxy cache. This assumes that you want to download *all* of the RPM files on the Satellite server to the Proxy. You need to run the cron job and **rsync** command as root on the Proxy server, but can then log in to the Satellite server as any user that has read access to the RPM store; this should be all users.

Configuring SSH Key Pairs to Enable Automatic Connection to the Satellite

You need to set up SSH key pairs that enable the **root** user on the Proxy server to establish an SSH connection to **$user** on the Satellite server, without typing a password.

Important

This method requires that the private key does not have a password. Ensure that you keep this key safe.

Run the following command as **root** on the Proxy server:

```
ssh-keygen -t rsa -N '' -f /root/.ssh/id_dsa \
&& cat /root/.ssh/id_dsa.pub | ssh user@satellite.example.com 'cat >>
.ssh/authorized_keys'
```

Enter the password for *user* when prompted. You should now be able to create an SSH connection to your Satellite server, without typing a password:

```
# ssh user@satellite.example.com
```

Setting up Automatic Synchronization

You now need to add a **cron** job with an appropriate **rsync** command. For example, as root on the Proxy:

```
crontab -e
1 3 * * * /usr/bin/rsync -r
user@satellite.example.com:/var/satellite/redhat/*/ /var/spool/rhn-
proxy/rhn
# write and quit with :wq
```

This configuration runs the **rsync** command at 3:01 a.m. every day and downloads any RPM files that are on the Satellite server into the Proxy cache. The **/var/satellite** directory is the default Satellite root directory, but this is configurable. If you have changed it on your Satellite installation then adjust this command appropriately.

Chapter 5. Configuring Satellite Proxy to Use CNAME Records

One of the features of Red Hat Satellite Proxy is the ability to take advantage of CNAME records, or aliases, instead of the canonical host name. This feature is useful if there are network issues that prevent communication with Red Hat Satellite Proxy using its normal host name or configuration.

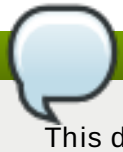
5.1. Prerequisites

To use CNAME records with your Red Hat Satellite Proxy Server installation, you need to:

- Ensure that the required CNAME records have been configured for the server.
- Add the CNAME records to the Satellite Proxy server configuration.
- Create a new multi-host certificate and install it on the Satellite Proxy.

5.2. Adding CNAME Records to the Satellite Proxy Server Configuration

This section describes how to add your Satellite Proxy server's CNAME records to the Satellite Proxy server configuration, and have those CNAMEs recognized as valid options within the Satellite Proxy web interface.



Note

This document does not cover how to set up DNS CNAME records for the machine where your Satellite Proxy server is installed. Contact your system or DNS administrator to ensure the required CNAME records are correctly set up.

Procedure 5.1. To Configure Your Satellite Proxy Server to Use CNAME Records:

1. Determine the system ID of your proxy server. You can find this value in the **`/etc/sysconfig/rhn/systemid`** file on your proxy server. Locate the following stanza in this file (your system ID will be different from this example):

   ```
   <member>
   <name>system_id</name>
   <value><string>ID-1036997498</string></value>
   </member>
   ```

   The system ID is contained within <string> tags. Do not include "ID-" as part of the actual system ID.

2. Add the following line to the **`/etc/rhn/rhn.conf`** file. Replace *systemID* with the value from the previous step:

   ```
   valid_cnames_sytemID= cname1,cname2,cname3
   ```

3. Run the following command to restart the Tomcat service:

   ```
   # service rhn-proxy restart
   ```

After the Tomcat service has restarted, refresh the Satellite Proxy server web interface and you should see the CNAMEs listed on the **Hardware** tab of the **System Details** page.

Before you can use these CNAMEs, however, you need to create a new set of certificates, and configure the Satellite Proxy to use these certificates. This is because the original certificate is only valid for the canonical host name; you need to create new certificates that are valid for each CNAME. This is covered in Section 5.3, “Generating and Using Multi-host SSL Certificates”.

Report a bug

5.3. Generating and Using Multi-host SSL Certificates

You need to generate multi-host SSL certificates to take advantage of the ability to use CNAME records on the Satellite Proxy server. You also need to update the **rhn-ca-openssl.cnf** file to ensure that the Satellite Proxy server is aware of and uses these certificates.

Procedure 5.2. To Update the SSL Configuration File to use Multi-host Certificates:

1. Edit the **/root/ssl-build/rhn-ca-openssl.cnf** file and locate the *[CA_default]* section.
2. Ensure the entry **copy_extensions = copy** exists and is not commented out.
3. Save and close the file.

Important

You need to complete the above step before you run **configure-proxy.sh** with **SSL_CNAME** set, or the installation will fail.

You also need to update your *answers* file so that the Satellite Proxy configuration will use the new SSL certificates created previously.

Procedure 5.3. To Update the Answers File to Use Multi-host SSL Certificates:

1. Edit the **answers.txt** file that you created for the initial Satellite Proxy installation. If you did not create such a file, you can find an example setup in **/usr/share/doc/spacewalk-setup-<version>/answers.txt.**
2. Ensure the following line exists, and is not commented out:

```
SSL_CNAME = (cname01 cname02 cname03)
```

3. Run the **configure-proxy.sh** script with the **--answer-file** option to generate the multi-host SSL certificate. For example:

```
# configure-proxy.sh --answer-file=</path/to/answers.txt>
```

Note

You can run the **`configure-proxy.sh`** script multiple times to test or update configurations, as required.

Report a bug

Chapter 6. Load Balancing Satellite Proxy Servers

Some environments include a load balancer between Satellite clients and Satellite proxies to distribute the load of Satellite requests, or to help redirect requests to a location closer to the originating client. If the Satellite Proxy topology becomes more complex, includes CNAME support, chaining, and so on, it is helpful to test the HTTP header requests exchanged between load balancers and round-robin proxy chains. This chapter describes configuring Squid as a reverse proxy to perform round-robin requests between two Satellite proxies. It covers the set up procedure and how to support both non-SSL and SSL proxy requests.

The following environment in this example would use five different hosts:

- Red Hat Satellite Proxy A, signified with IP address **192.168.100.16** and hostname **proxya.example.com**
- Red Hat Satellite Proxy B, signified with IP address **192.168.100.17** and hostname **proxyb.example.com**
- Load Balancer, signified with hostname **lb.example.com**
- Red Hat Satellite Server with Red Hat Satellite Proxy A and B connected
- The client machine, signified with IP address **192.168.100.19**

6.1. Installing a Squid Reverse Proxy

Install a Squid server to use as the load balancer by using reverse proxy mode.

```
# yum install squid
```

You also need to generate SSL certificates and sign them with the Satellite CA. The easiest method is to use the **rhn-ssl-tool** on the Satellite server to generate the server certificates, because the CA is already available.

The Satellite SSL Maintenance Tool (rhn-ssl-tool) generates and maintains Satellite SSL keys and certificates. It also generates RPMs for use in deploying these keys and certificates. The tool is geared for use in a Satellite context, but can be useful outside of Satellite too.

In this example, the load balancer is called **lb.example.com**; substitute the host name that applies to your deployment, and enter a suitable build directory. Run this command on the Satellite server.

```
$ rhn-ssl-tool --gen-server --set-hostname=lb.example.com -d /root/ssl-
build
```

The **rhn-ssl-tool** used above creates SSL files for **lb.example.com** and saves the files in **/root/ssl-build** directory. Copy the **server.crt**, **server.key**, and the **RHN-ORG-TRUSTED-SSL-CERT** CA certificate from the **dhcp** directory to the **lb.example.com** load balancer. These files are used to set up SSL for the actual load balancer. The **RHN-ORG-TRUSTED-SSL-CERT** certificate allows SSL communication between the load balancer and the proxies.

Modify the **/etc/squid/squid.conf** file on the **lb.example.com** server to set up reverse proxy mode:

Example 6.1. Setting up Reverse Proxy Mode

```
#
# SSL configuration
#

# Ensure you enter each configuration directive on a single line

acl is_ssl port 443

https_port 443 cert=/etc/pki/tls/certs/lb.crt
key=/etc/pki/tls/certs/lb.key accel vhost name=proxy_ssl

cache_peer proxya.example.com parent 443 0 no-query originserver
round-robin ssl name=proxya.example.com
sslcafile=/etc/pki/tls/certs/squid-ca.crt

cache_peer proxyb.example.com parent 443 0 no-query originserver
round-robin ssl name=proxyb.example.com
sslcafile=/etc/pki/tls/certs/squid-ca.crt

cache_peer_access proxya.example.com allow is_ssl
cache_peer_access proxya.example.com deny !is_ssl
cache_peer_access proxyb.example.com allow is_ssl
cache_peer_access proxyb.example.com deny !is_ssl

#
# Non-SSL configuration
#

# Ensure you enter each configuration directive on a single line

acl nonssl port 80

http_port 80 accel name=proxy_nonssl defaultsite=dhcp16.example.com

cache_peer 192.168.100.16 parent 80 0 no-query name=proxy_nonssl
originserver

cache_peer_access proxy_nonssl allow nonssl
cache_peer_access proxy_nonssl deny !nonssl

sslpassword_program /bin/password.out
forwarded_for on
```

The previous example demonstrates setting up two reverse proxies. Port 443 has two proxies that are used in round-robin mode. Requests are shared equally between the two proxies. The **server.crt** and **server.key** files were renamed to **lb.crt** and **lb.key** respectively (short for load balancer) for easier identification. The Satellite CA certificate was renamed to **squid-ca.crt**; the **cache_peer sslcafile** option refers to this file.

Add the certificates to the **squid** group:

```
# chgrp squid /etc/pki/tls/certs/{lb.crt,lb.key,squid-ca.crt}
```

The file details should appear as follows:

```
-rw-r--r--. 1 root squid   5450 Aug 23 21:23 lb.crt
-rw-r--r--. 1 root squid   1675 Aug 23 21:23 lb.key
-rw-r--r--. 1 root squid   5363 Aug 22 14:19 squid-ca.crt
```

The **cache_peer** directives set up the two proxies that will be used in round-robin format. Note that you need to specify the CA certificate so that the load balancer can communicate with the proxies. Further, we are only allowing port 443 traffic to hit these proxies using the squid **acl is_ssl** and **cache_peer** directives.

All traffic on port 80 is redirected to one proxy and defaults to the dhcp16.example.com proxy using the **defaultsite** directive. Acls are set up similar to the ssl port.

The **sslpassword_program** directive allows you to send the SSL key passphrase (if used; displayed for completeness) to squid on startup without human intervention. The contents of password.out is a bash script that echos the SSL passphrase. The **forwarded_for** directive configures the load balancer to send the **forwarded_for** headers to the proxies.

Important

Edit the **/etc/squid/squid.conf** and comment out the default port, 3128, that squid normally listens on:

```
# Squid normally listens to port 3128
# http_port 3128
```

Restart squid after config modifications:

```
# service squid restart
```

6.2. Setting up the Client

You need to modify the **/etc/sysconfig/rhn/up2date** file on the client to correctly address the load balancer:

```
serverURL[comment]=Remote server URL (use FQDN)
serverURL=https://lb.example.com/XMLRPC
```

The load balancer uses the Satellite CA certificate for its signed SSL certificate; you do not need to modify the client CA certificate values.

6.3. Testing the Configuration

This section discusses basic testing of your load balancer and proxy configuration.

Initial Testing

Start the load balancer and the proxy machines. Note that the Satellite Proxy logs will not be created until the first requests are delivered (**/var/log/rhn/**). Try to install and remove an RPM file, such as zsh.

The **/var/log/squid/access.log** file on the load balancer should contain information similar to the following:

```
1377540630.159
97 192.168.100.19 TCP_MISS/200 1515 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxya.example.com text/base64
1377540630.733
529 192.168.100.19 TCP_MISS/200 1409 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxyb.example.com text/xml
1377540639.968
87 192.168.100.19 TCP_MISS/200 3742 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxya.example.com text/xml
1377540644.273
83 192.168.100.19 TCP_MEM_HIT/200 2238956 GET
https://lb.example.com/XMLRPC/GET-REQ/rhel-x86_64-server-
6/getPackage/zsh-4.3.10-5.el6.x86_64.rpm - NONE/- application/octet-
stream
1377540646.765
100 192.168.100.19 TCP_MISS/200 1515 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxyb.example.com text/base64
1377540647.291
485 192.168.100.19 TCP_MISS/200 1409 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxya.example.com text/xml
```

This example shows that the client requests are balanced between the two different proxies. A single **yum** package install is still shared between the multiple proxies because multiple requests are involved during a package installation. The client IP address is 192.168.100.19 and you can see the named **proxya.example.com** and **proxyb.example.com** cache peers corresponding to the two different Satellite Proxies.

Squid Load Balancer Log Files

Refer to the following log files to monitor load balancer activity:

- **/var/log/squid/access.log** for watching requests arriving from the client
- **/var/log/squid/cache.log** for debugging SSL startup issues and cache peers on ports 80 and 443

You can use the **netstat** command to ensure that the squid load balancer is listening on the correct ports:

```
# netstat -tulpn | grep squid | grep tcp
tcp        0      0 :::80                   :::*                    LISTEN
29120/(squid)
tcp        0      0 :::443                  :::*                    LISTEN
29120/(squid)
```

Chapter 7. Upgrading a Red Hat Proxy Server Installation

This chapter describes how to upgrade your Proxy Server installation. These instructions assume that you have a fully functional Proxy Server and its required entitlements.

7.1. Prerequisites

The latest version of Red Hat Satellite Proxy Server requires:

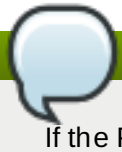
- Red Hat Enterprise Linux 6 (64-bit only).
- Removal of the previous Proxy server's system profile from the parent Satellite server.

7.2. Upgrading Your Proxy Installation

1. Back up your existing Proxy server. If applicable, restore the SSL build directory from the backup to the directory **/root/ssl-build**.
2. Register the Proxy to the parent Satellite. Make sure that the Proxy is subscribed to both the Red Hat Enterprise Linux Server base channel and the Red Hat Network Tools child channel.
3. Install the **spacewalk-proxy-installer** package from the Red Hat Network Tools child channel:

```
# yum install spacewalk-proxy-installer
```

4. Install the latest version of Proxy, as documented in Section 3.5, “Installing the Red Hat Satellite Proxy Server Packages”.



Note

If the Proxy server is registered to Red Hat Satellite, and the Proxy previously managed custom channels, restore the custom package repository from the pre-upgrade backup. The permissions and ownership will also need to be set up properly.

```
# chmod 0750 /var/spool/rhn-proxy
# chown apache:apache /var/spool/rhn-proxy
# mkdir -m 0750 -p /var/spool/rhn-proxy/list
# chown apache:apache /var/spool/rhn-proxy/list
```

The default custom package repository is usually **/var/spool/rhn-proxy**.

5. After the installation, update the server to the latest errata updates:

```
# yum update
```

6. Restart the Proxy Server services and test the Proxy Server's functionality:

```
# /usr/sbin/rhn-proxy restart
```

Report a bug

Sample Satellite Proxy Server Configuration File

The **/etc/rhn/rhn.conf** configuration file for the Satellite Proxy provides a means for administrators to establish key settings. Take care when making any changes, however, because any configuration errors in this file may cause Satellite Proxy failures.

Pay close attention to the **traceback_mail** and **proxy.rhn_parent** parameters. Review the sample and its comments (comments are identified by a hash mark (#) in column 1), for additional details.

Important

You can add the **use_ssl** option to **rhn.conf** for testing purposes. Set its value to 0 to turn off SSL between the Satellite Proxy and the Satellite server. Be aware that this greatly compromises security. Reset this option to its default value of 1 to re-enable SSL, or remove the line from the configuration file.

```
# Automatically generated RHN Management Proxy Server configuration
file.
# ------------------------------------------------------------------------
---

# SSL CA certificate location
proxy.ca_chain = /usr/share/rhn/RHNS-CA-CERT

# Corporate HTTP proxy, format: corp_gateway.example.com:8080
proxy.http_proxy =

# Password for that corporate HTTP proxy
proxy.http_proxy_password =

# Username for that corporate HTTP proxy
proxy.http_proxy_username =

# Location of locally built, custom packages
proxy.pkg_dir = /var/spool/rhn-proxy

# Hostname of RHN Server or RHN Satellite
proxy.rhn_parent = rhn.redhat.com

# Destination of all tracebacks, etc.
traceback_mail = user0@domain.com, user1@domain.com
```

Report a bug

Glossary of Terms

To better understand Red Hat Satellite Proxy, it is important to become familiar with the following Red Hat Satellite terms:

Channel

A channel is a list of software packages. There are two types of channels: base channels and child channels. A *base channel* consists of a list of packages based on a specific architecture and Red Hat release. A *child channel* is a channel associated with a base channel that contains extra packages.

Organization Administrator

An Organization Administrator is a user role with the highest level of control over an organization's Red Hat Satellite account. Members with this role can add and remove other users, other systems, and system groups to the organization. Each Red Hat Satellite organization requires at least one Organization Administrator.

Channel Administrator

A Channel Administrator is a user role with full access to channel management capabilities. Channel Administrators can create channels and assign packages to channels. Organization Administrators can use the **Users** tab on the Red Hat Satellite website to assign this role to other users.

Red Hat Update Agent

The **Red Hat Update Agent** is the client application that allows users to retrieve and install new or updated packages on the client system.

Traceback

A traceback is a detailed error description that is useful for troubleshooting the Satellite Proxy. Traceback files are automatically generated when a critical error occurs, and they are then emailed to the individuals designated in the Satellite Proxy's configuration file.

See the *Red Hat Reference Guide* and the **Help** page on the Satellite Web user interface for more information.

Revision History

Revision 4-9 **Mon Apr 20 2015** **Dan Macpherson**
Fixing typo in load balancing section

Revision 4-8 **Thu Feb 19 2015** **Dan Macpherson**
Removing unnecessary channel

Revision 4-7 **Tue Feb 17 2015** **Dan Macpherson**
Minor Maintenance updates

Revision 4-6 **Tue Feb 3 2015** **Dan Macpherson**
Pushing maintenance update for Satellite 5.7

Revision 4-5 **Wed Jan 7 2015** **Dan Macpherson**
Packaging snapshot versions

Revision 4-4 **Thu Jan 1 2015** **Dan Macpherson**
Release Candidate for Satellite 5.7

Revision 4-3 **Mon Dec 8 2014** **Dan Macpherson**
Preparing books for technical review

Revision 4-2 **Wed Dec 3 2014** **Dan Macpherson**
Completed major revision of the book and tested end-to-end installation workflow

Revision 4-1 **Fri Nov 21 2014** **Dan Macpherson**
Revised Software Requirements section

Revision 4-0 **Mon Nov 3 2014** **David O'Brien**
Add section "Summary of Steps" to cover Proxy installation.
BZ 1155876 - Fix incorrect and inconsistent units for h/w requirements.
BZ 1154963 - Add chapter on Proxy precache.
BZ 1112093 - Update list of supported hypervisors.
BZ 1145826 - iptables settings incorrect for proxy.

Revision 3-23 **Fri Sep 27 2013** **Dan Macpherson**
Final version of documentation suite

Revision 3-22 **Thu Sep 12 2013** **Dan Macpherson**
Minor changes

Revision 3-21 **Wed Sep 11 2013** **Dan Macpherson**
Implemented QE feedback for BZ#1001363

Revision 3-20 **Wed Sep 11 2013** **Dan Macpherson**
Added initial steps for Proxy Installation as per BZ#1001360

Revision 3-19 **Wed Sep 11 2013** **Dan Macpherson**
Change to Product and BookID

Revision 3-18 **Wed Sep 11 2013** **Dan Macpherson**
Updating requirements

Revision 3-17 **Tue Sep 10 2013** **Dan Macpherson**
Revised Subtitle, Abstract and Preface for all Guides

Revision 3-16 **Thu Aug 29 2013** **Dan Macpherson**
First implementation of QE Review feedback

Revision 3-15 **Sun Jul 28 2013** **Dan Macpherson**
Removing references to Red Hat Network usage of Proxy as per Technical Review

Revision 3-14 **Sun Jul 28 2013** **Dan Macpherson**
Second implementation of tech review feedback

Revision 3-13 **Wed Jul 24 2013** **Dan Macpherson**
Corrections for BZ#987245

Revision 3-12 **Tue Jul 23 2013** **Dan Macpherson**
First implementation of tech review feedback

Revision 3-11 **Fri Jul 12 2013** **Dan Macpherson**
Final beta updates

Revision 3-10 **Fri Jul 12 2013** **Dan Macpherson**
Beta documentation update

Revision 3-9 **Fri Jul 12 2013** **Dan Macpherson**
Beta documentation update

Revision 3-8 **Thu Jul 11 2013** **Dan Macpherson**
Beta documentation creation

Revision 3-7 **Thu Jul 11 2013** **Megan Lewis**
Updated preface.
Updated section on base installation.

Revision 3-6 **Wed Jun 26 2013** **David O'Brien**
Updated requirements.
Updated section on how Proxy Server works.
Updated graphics for section on topologies.
Major update to section on installing Proxy Server.
Add section on using cnames.

Revision 3-5 **Wed Sep 19 2012** **Dan Macpherson**
Final packaging for 5.5

Revision 3-4 **Wed Jul 4 2012** **Athene Chan**
Prepared for 5.5 release
Technical review changes incorporated
BZ#491007 Added Chapter on Upgrade Install

Revision 3-0 **Wed Jul 4 2012** **Athene Chan**

Prepared for 5.5 release
Technical review changes incorporated
BZ#491007 Added Chapter on Upgrade Install

Revision 2-5 **Thu Jan 5 2012** **Lana Brindley**

BZ#682996 - Installation chapter - Update instructions
BZ#705755 - Package manager chapter - Additional info
BZ#722193 - Requirements chapter - Fixed error
BZ#729617 - Installation chapter - Fixed error
BZ#729663 - Installation chapter - Added admonition

Revision 2-4 **Mon Aug 15 2011** **Lana Brindley**

Folded z-stream release into y-stream

Revision 2-3 **Wed Jun 22 2011** **Lana Brindley**

BZ#713527 - Added RHEL 6 references

Revision 2-2 **Wed Jun 15 2011** **Lana Brindley**

Prepared for publication

Revision 2-1 **Fri May 27 2011** **Lana Brindley**

Updates from translators

Revision 2-0 **Fri May 6 2011** **Lana Brindley**

Prepared for translation

Revision 1-9 **Wed April 27 2011** **Lana Brindley**

BZ#653844 - QE Review

Revision 1-8 **Mon Feb 7 2011** **Lana Brindley**

BZ#646176 - Installation